DRA

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

March 1983 - August 1983

William B. Rouse, Principal Investigator
John M. Hammer
Nancy M. Morris
Edward N. Brown
Wan C. Yoon

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

(NASA-CR-173782) PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS Semiannual Progress Report, Mar. - Aug. 1983 (Georgia Inst. of Tech.) 73 p HC A04/MF A01 CSCL 05H G3/54

N84-29464
Unclas
17784

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Further, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering methods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments as affected by the human's level of knowledge of system operations. Both of these efforts are producing results that are planned to be tested further in the Center's new full-scale simulation facility. Progress on developing this facility is discussed in the third and final section of this progress report.

THE INTELLIGENT COCKPIT

This project is a direct descendent of work by the authors on human-computer interaction in the cockpit dating back to 1975. As this research has evolved, the modeling and analysis methods that have emerged have enabled consideration of increasingly complex domains. For example, two of the more recent sets of studies considered pilot (and crew) problem solving in full-mission simulation studies [Rouse et al., 1982; Johannsen and Rouse, 1983].

The perspectives provided by these years of research have resulted in an integrated computer aiding concept which the authors have termed the "intelligent cockpit." The overall outline of this concept is outlined in Hammer and Rouse [1982]. The basic idea is to use advanced software engineering methods (e.g., from artificial intelligence) and models of human decision making and problem solving to produce a computer-based aid that "understands" what is going on in the cockpit and can provide assistance accordingly.

This very ambitious project is being pursued in an incremental manner. The first increment is an intelligent flight management aid that understands the nature of procedures and can monitor their execution. The paper by John Hammer in the Appendix of this report summarizes this work. The results reported in this paper prove the soundness of the concepts; the next stage will be to implement this idea in the Center's simulator to allow full-scale testing.

References

1. Hammer, J.M. and Rouse, W.B. 1982. "Design of An Intelligent Computer-Aided Cockpit." Proceedings of the 1982 IEEE International Conference on Cybernetics and Society, Seattle, October, pp. 449-453.

2. Johannsen, G. and Rouse, W.B. 1983. "Studies of Planning Behavior of Aircraft Pilots in Normal, Abnormal, and Emergency Situations," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, May/June 1983, pp. 267-278.

3. Rouse, S.H., Rouse, W.B. and Hammer, J.M. 1982. "Design and Evaluation of An Onboard Computer-Based Information System for Aircraft." IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 4, July/August 1982, pp. 451-463.

STUDIES OF HUMAN PROBLEM SOLVING

In order to support domain-oriented projects such as the intelligent cockpit. it is necessary to increase our basic understanding of human decision making and problem solving. This has been a main tenet of this program of research since its inception and continues to be a guiding principle.

The latest efforts in the area of basic studies of human problem solving have focused on the question of what humans need to know about a dynamic process in order to be able to deal successfully with unfamiliar and unanticipated events. The paper by Nancy Morris in the Appendix of this report summarizes the results of a study that compared knowledge of operating procedures and physical principles in a process control task. One of the most interesting results was that knowledge of physical principles, as assessed via a written test, did not result in improved performance. Of the many important issues that this raises, two of particular note are the nature of training (e.g., "what" vs. "how") and the appropriate forms for different types of knowledge.

FLIGHT SIMULATION SOFTWARE

The progress report for the last reporting period discussed the hardware modifications planned for the Center's DC-8 simulator in order to create an advanced cockpit simulator. This section focuses on software developments.

Software has been developed to produce a B-747 flight simulation on the Center's Vax 11/780. At this point in time, it employs simplified dynamics to simulate the motion of an aircraft and only a few of the essential subsystems. Despite this simplicity, the software meets our overall need to provide pilots with a reasonably realistic environment for the purpose of investigating their problem solving behavior in various situations.

The program allows the pilot to activate the control surfaces of the jet aircraft, adjust engine thrust, and tune navigational radio equipment. The program responds to commands by adjusting aircraft attitude to match the control surfaces and updating the instrument panel display as the trajectory of the aircraft evolves through space.

An instrument panel was designed to display information that comes from the flight simulation. This information is composed of current aircraft attitude, positions of switches, flight situation, and navigational environment. Included are pitch, altitude, engine thrust, compass, fuel, landing gear, brake, VOR system, stall warning, VLF OMEGA, ILS, VHF channel, etc. This brief panel gives the pilot all the basic flight information he needs during the three primary mission phases (i.e., takeoff, navigation, and landing) using standard flight procedures and radio facilities.

Although the pilot completely controls the motion of the jet, wind forces that vary with altitude can influence the flight. An analytical combination of jet and wind motion yields the true position of the jet relative to the earth's surface.

This simulation software, however, is still incomplete. The current interface--keyboard and screen--are suitable for software development but will have to change to use the simulator displays and controls. The existing single instrument panel must be rearranged into several different CRT's. The flight control will be executed by a pilot who will be sitting down in a full scale aircraft cockpit facility and using a control stick and a flight deck of high fidelity.

Also, more subsystems will be involved to cover all information that would be necessary for a realistic problem solving environment. Among these subsystems are the engines (giving engine pressure ratio, fuel flow, exhaust gas temperature, etc.), hydraulic system, autopilot, fuel, CDTI, etc. With all these subsystems, it is believed that the flight simulator will be an appropriate base for studies of aiding problem solving.

APPENDIX

AN INTELLIGENT FLIGHT MANAGEMENT AID FOR PROCEDURE EXECUTION

John M. Hammer

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

A computer program is described which contains a model of the procedures used in the operation of a twin engine aircraft. This program, by comparing the model to the aircraft state, can determine when a procedure (or checklist) should be or is invoked and when each step (detectable by a change in the aircraft state) is completed. Thus, the program tracks the flight crew's procedure execution through changes in the aircraft state. Data was used for evaluation from an earlier experiment on a Link GAT-II simulator. The program was able to identify practically all of the errors identified by hand as well as locate some that had been missed by human judges. It is felt that this model could signficantly aid flight crews.

INTRODUCTION

A computer program for detecting pilot error is described. This program observes pilot actions through the aircraft controls and state. These actions are compared to those of a procedure script, which can be considered a prescriptive model of the procedural aspects of flight. A pilot error, then, is a discrepancy between the pilot actions and the script. The program is capable of detecting omitted, incorrect, or out-of-order steps as well as certain irrelevant actions.

This procedural aid is part of a larger research thrust known as the intelligent cockpit. Its goals are to demonstrate concepts for a system capable of intelligent decision-aiding in flight management. For example, Hammer and Rouse [1982] have identified a number of levels at which aiding could occur. At the lowest level, computerized warning, calculation, and display control could, if implemented properly, improve the human factors of the cockpit. Most flight deck automation is concerned with this level [Wiener and Curry 1980]. At a second level, the computer could check that certain conditions were met and infer the intentions of the flight crew. For example, the system described later checks pilot actions against a prescribed plan and infers what procedure the crew should be following. At the highest level, the computer could compensate for and advise the pilot. Compensation could mean taking over some task that had been allocated to the pilot or correcting for pilot error. Advice could take the form of a natural language dialogue on some cooperative human-computer problem solving. Both of these forms of aiding are beyond the current state of the art.

Problem Statement

Currently, pilot error (that goes undetected by the pilot) is found by humans who examine simulator traces (as above) or cockpit voice recordings. It would be better to detect these errors seconds after they occurred while there was still time to correct them. This was the goal of the research reported here. Software was written to implement a model of procedure execution during flight. The model is continually updated (steps noted done, procedures invoked and finished) during the flight so as to keep it close to what the flight crew is doing. The model could be used as an aid since it has some understanding of when a procedure is to be used and what its execution entails. To evaluate the model, it was tested on earlier simulator flights with previously indentified errors. The figure of merit was the number of these errors that the model could locate.

The remainder of this article contains sections on previous work on procedural error, programming methodologies for procedural aiding, the pilot's procedures, the internal program organization of the aid, evaluations of the aid, and conclusions.

PREVIOUS WORK

Humans occasionally err when following procedures. The forms of error have frequently been observed to be steps not executed, done out-of-order, done incompletely, or done at the wrong time. The same is true of whole procedures, which are sometimes incorrectly selected for execution. Some theories [Norman 1981] and many classification schemes [Rasmussen 1979] [Monan 1979] [van Eekhout and Rouse 1981] have been put forth, and are reviewed in Rouse and Rouse [1983]. The theories and classification schemes will not be reviewed here, since the goal is to build an aid for reducing error, not to explain it.

The remainder of this literature review contains two parts: one on other aids for reducing procedural error, and a second on a line of research leading directly to this research.

Goodstein [1979] has proposed a computerized procedure display system. Its design was based on the belief that the operator executes procedures with some goal in mind -- changing the system via procedures from one state to another. Consequently, the system explicitly displays this hierarchy. The procedure environment is also enriched by including preconditions, constraints, and warnings along with the procedure text.

The proposed system was to be implemented with three displays. The first would display the sequence of procedures to be followed. Included in this would be the status of various procedures (e.g., on hold or waiting for the plant to respond). The second display would contain the text of a single procedure along with supplemental preconditions, constraints, and warnings. The third display is for support. It might display the relevant plant status so that the operator would not have to walk away from the displays just to read an instrument. While this proposed three display system would seem to be an improvement over current practice, it does not appear to have been evaluated with human operators.

Colley [1982] and Seeman et al. [1982] are in the process of developing a computerized procedure support system for nuclear power plant operators. The system compares the current plant state to a set of desirable, or safe, plant states. A procedure is then generated by the computer to transform the plant to the nearest safe state. A practical advantage of automatic procedure generation is that a potentially larger set of procedures could be available than would from hardcopy. For the latter, the system designers cannot afford to create every possible procedure. If a computer could derive the procedures from some general principles, automatic generation could represent a considerable improvement.

The current system can produce procedures for an eighteen component lubrication system. The procedures are generated dynamically; i.e., after each operator action, the system regenerates an appropriate procedure. Thus, if the operator errs, an appropriate change will appear in the procedure text. The development effort should be viewed as an attempt to produce a methodology for procedure generation. It has not yet been tested on operators.

Background Work Leading Directly to this Study

This section discusses a sequence of studies that lead to the work presented in this article. Rouse and Rouse [1980] studied displays for procedures in an abstract scenario. Three displays were used: a traditional hardcopy, a practically identical softcopy (displayed on a CRT screen), and a cued softcopy that dimmed a procedure step when it had been completed. To simulate the distractions faced by pilots, an arithmetic side task was added. The experimental results showed the cued softcopy display to be significantly faster and to cause fewer errors than the other two displays.

In a second, similar experiment conducted in a realistic environment, Rouse, Rouse, and Hammer [1982] studied hardcopy and cued procedure displays in a Link GAT-II twin engine aircraft simulator. Their experiment will be rather carefully described because the data from it was reanalyzed for the research reported here.

The aircraft simulator was configured as a Piper Aztec F. A PDP-11/40 minicomputer was interfaced to the simulator and recorded timestamped changes of the aircraft state. The record of a flight, termed a simulator trace, consisted of a sequen\e of triples, where each triple contained a time, a signal, and a signal value. A signal was recorded only at a significant change, which usually was a deviation of ±10% from its previously recorded value. (It is this data that was analyzed in the research reported here.) Also interfaced was a special purpose keyboard that controlled the CRT procedure display, one level of an independent, display variable. The other level was a traditional, printed hardcopy procedure.

Subjects flew in normal, emergency, and double emergency flights. The eight subjects were all instrument- and twin engine-rated pilots with the exception of one who had 70 hours of twin engine simulator time and was judged to have been equal to the others. Subjects flew in 3 flight scenarios. The normal flight was a departure, climb to 2000 feet, direct cruise to another airport, descend, and land. The emergency flight was a single engine failure after the aircraft climbed and through 2000 feet. The double emergency failure consisted of a single engine failure at the same point plus a gear extension failure during the single engine pre-landing procedure.

The data analysis showed that the hardcopy procedure was 19% faster than the CRT display (statistically significant at p<.025). The CRT display produced 7.5 times fewer errors of the kind that could possibly be affected by display (p<.025).

A finer grained analysis of the experimental data from the above experiment appears in [Rouse and Rouse, 1983]. Forty-three errors were classified as occurring during hypothesis check, goal choice, procedure choice, or procedure execution. Errors were also classified as incorrect or unnecessary across all of these four categories. Displays were found to have a significant effect on errors that were categorized as wrong or incorrect. No effect was seen on errors classified as unnecessary actions.

PROGRAMMING METHODOLOGIES

The most appropriate programming methodology depends on the type of problem to be solved. This is true even though all methodologies are theoretically equal, since humans may find certain programs easier to express in one methodology than another. Though many methodologies exist, only two will be discussed here: conventional von Neumann programming [Backus 1978] and symbolic programming.

Conventional programming is often represented by the typical FORTRAN, BASIC, COBOL, PL/I, and Pascal program. Each computational step has one or more input values (or vectors) and produces a single output value (or vector). The values are usually numbers or characters. Such a methodology is best suited for numeric or data processing tasks such as aircraft simulator dynamics or implementing the lowest level of aiding: warning, calculation, and display control.

Symbolic computation, done primarily in Lisp or perhaps Prolog, is better suited to higher levels of aiding because the problem the human solves is itself symbolic. In other words, an aid should use symbolic computing to solve symbolic problems. The two methodologies employed are rule-based systems and scripts.

Rule-Based System

A rule-based system (RBS) [Waterman and Hayes-Roth, 1978] is one form of symbolic computation often used for an expert system, a program capable of rivaling human performance in a small but complex problem domain. Some examples of expert systems are:

1. MYCIN [Shortliffe, 1976] - selects antimicrobial therapy for infections.
2. DENDRAL [Buchanan et al., 1969] - analyzes mass spectroscopy data to reconstruct the original molecule from its constituents.
3. PONTIUS-0 [Goldstein and Grimson, 1977] is a system that achieves attitude instrument flight.
4. Wesson [1977] has produced a program to perform the enroute ATC function with performance (under real world conditions) as good as a human controller.

The structure of a RBS contains two principal parts: working memory and rules. For flight management, working memory can be assumed to contain the entire state of the aircraft (e.g. airspeed, altitude, pitch, roll, engine variables, electrical variables) as well as additional temporary memory. A rule contains two parts: a situation (such as altitude decreasing or airspeed > Vx) and an action (such as a procedure or storing some value in working memory). The following example shows possible rules for the pilot's handling engine failure during takeoff:

RULE	SITUATION	ACTION
1.	airspeed < Vmc	close throttles stop on runway
2.	Vmc < airspeed < Vx	abort flight close throttles stop on runway
3.	VMC < airspeed < Vx and sufficient runway	accelerate to Vx
4.	Vx < airspeed	maintain control and speed clean up aircraft climb secure engine land as soon as possible

The rules operate as follows. If the airspeed ≤ VMC when one engine fails (the aircraft is in contact with the runway), the situation of rule 1 applies, and the flight is aborted as per the action of rule 1. Rule 3 gives a further example of how rules are invoked. If Rule 3 is applied, airspeed will be increased to at least Vx, at which time Rule 4 applies. Thus, one rule may transfer control to another rule either by a change in the aircraft state as in this example or in temporary memory (not illustrated).

In the system discussed here, rules are used primarily for their ability to recognize situations. In other words, rules detect pilot actions and changes in the aircraft state (e.g., landing gear up) that indicate a new mode of operation (e.g., from on the ground to airborne). The rules, however, are not self-organized; they are held together by scripts.

Scripts

The script [Schank and Abelson 1977], the final programming methodology discussed here, is a form of symbolic computation like rule-based systems. Where a RBS recognizes specific situations and invokes the corresponding actions, a script describes the expected actors and actions in some situation. The script is a construct similar to the frame [Minsky, 1975] and to schema or template [Bartlett, 1932].

As an example of how the script concept might apply, consider a script for landing an airplane. The landing script provides the desired aircraft configuration -- engine settings, flaps, gear -- and their changes over time. Some of these will be dependent on the airport, and hence the landing script will have airport-dependent parameters. In addition, the landing script will indicate the scripts most likely to be activated next -- taxi, go-around, travel to an alternate airport, etc.

The power of scripts comes both from their rich description of actions and from the ability to determine which script is really active. The original application of scripts was understanding natural language (e.g., English). In spoken language, the speaker will, in the interest of economy, omit many details that the listener can infer. A script provides background for the computer so that it might draw some of the same inferences that a human would. To determine the next active script is a matter of selecting the script that best matches the current facts.

In a similar way, scripts can be used to infer what the flight crew is doing. The various controls and switch settings, sensed by the computer, can be viewed as a stream of details that partially conveys the flight crew's current thinking. By using scripts, the computer should be able to infer the full details in much the same way as it is used to understand natural language. In fact, one can envision an "advanced" intelligent cockpit where the computer would use the crew's conversation as one of its data sources. Though this may seem far-fetched, it will be demonstrated later that some errors could only be detected by this means.

PROCEDURES

Because the procedures pilots followed are central to this work, an example is given in Figure 1 of a typical procedure. Some aspects of these procedures will now be given. First, note that most steps are quite simple, e.g., steps 1 and 2; and the program senses their completion by a simple examination of the simulator state. Second, some steps cannot be sensed because the required signals are not available to the computer. An example is step 14; instrument vacuum was not recorded. Such steps are ignored by the program (deleted from the internal model at startup). Third, some steps call for the pilot to check a sensor reading. The program can check the sensor, but it cannot be sure the pilot has done so if the sensor reading is acceptable. Steps 10 through 13 are an example. Fourth, sensing some conditions may be difficult because the changes were not logged in the simulator trace because they were too small. For example, step 7 requires a 175 rpm drop, which is about 8% of the existing 2300 rpm. This change was unlikely to have been recorded in this data. Thus, the program can observe the magneto grounding but not its effect on engine rpm. This same problem also occurs when the pilot fine tunes the engines (leans mixture, changes propeller), as these changes are typically too small to be recorded. Of course, the problem of unavailable data would not be a problem in a real aircraft or in simulator data collection with a high sampling rate.

Some aspects of the simulator and aircraft in general make the sensing of steps more complex than it would first appear. First, some changes require time to occur. For example, in step 5, the propeller feather switch, which is discrete, may precede by a second the actual change of the propeller. A second difficulty is properly sensing temporary states. Two steps in the shutdown procedure, not shown, are an example. One is a momentary interruption of the magnetos and the other is a complete shutdown to stop the engines. Sensing the former requires that the transitions from ON to OFF to ON be observed within a short time frame. If this were not done, the program might misinterpret some other change to the magnetos.

INTERNAL PROGRAM STRUCTURE

The internal program structure will first be described in terms of a single procedure step. Next, the hierarchical organization of steps, procedures, and flight phases is described. Finally, the control structure, which interprets the steps, procedures, and phases, is described.

The first data structure is the aircraft database, which contains roughly seventy discrete and continuous signals. Each input record contains three items: a timestamp, a signal number, and a signal value*. As the input is read, new values are inserted in the database. Old values are not, however, immediately forgotten. Instead, they are retained if they are less than 6 minutes old or less than 100 in number so that the program may inspect earlier states.

The second data structure is the internal model of the procedures used by the crew. An individual procedure step (and other entities to be discussed later) is represented by the Pascal record shown in Figure 2. NAME is a text string that is used for humans to read. CAN_EXEC, DONE_EXEC, and ABORT_EXEC are rules (expressions that evaluate either true or false) that determine whether a step's STATE is considered UNSTARTED, IN_PROGRESS, DONE, or ABORTED according to the transition diagram shown in Figure 3. For example, for step 1 of the engine start procedure, the DONE_EXEC rule would check to see that both right and left mixture controls are currently at the full-rich setting.

ALLOWED et alia are sets of signals that can or cannot change during the execution of this step. These sets are used to detect actions that should not occur. When a simulator state change is read, these sets are examined to determine if the

*Other inputs -- keyboard entries, flight observer signals, etc. -- were ignored.

change is allowed. Only steps that are IN_PROGRESS are examined. Six sets were found necessary to detect pilot error. Normally, the program examines ALLOWED and DISALLOWED to determine the allowability of the signal. In engine-out emergency procedures, steps often refer to controls on the operative or inoperative engine. Thus, four more sets are necessary for the product of operative/inoperative with allowed/disallowed.

Procedure Step Hierarchy

Up to here, procedure steps have been described without mentioning their surrounding context. In fact, there is a hierarchy of four levels, with procedure steps at the lowest level. A number of steps are collected under a single procedure. One or more procedures are collected under a phase (e.g., pre-flight, takeoff), and all phases are collected under a single entity FLIGHT. Figure 4 illustrates the hierarchy. Each circle in Figure 4 corresponds to one script record as shown in Figure 2. Thus, all levels are represented uniformly. PARENT and COMP fields are used to represent the hierarchy.

The checkoff of procedures and phases is handled just as it is for procedure steps. There is some structure imposed on this process by the hierarchy, however. Only when a procedure STATE is IN_PROGRESS will its procedure steps (its subcomponents) be examined for transitions to new states. Further, when a procedure is DONE or ABORTED, its steps are not examined for transition. The structure imposes a preferred order of left-to-right on the execution of sub-scripts beneath a given script. The program expects execution in this order, but is capable of following any order. The program continually examines the DONE rules of all steps beneath an IN_PROGRESS procedure. The changing of the simulator state causes the rules to evaluate true in the order that the steps are completed.

The hierarchy also controls testing for allowed changes. First, only IN_PROGRESS steps, procedures, and phases are examined. All of the IN_PROGRESS steps are tested to determine if the signal is in one of the sets. If not, the same tests are made of IN_PROGRESS procedures, and, if necessary, of the IN_PROGRESS phases.

Emergency Daemon Procedures and Substitute Procedures

For the normal flight, the procedural hierarchy works well. During an emergency, flight operations are less structured. For example, a single engine-out emergency can happen any time the engines are running and the aircraft is airborne. Consequently, the procedure(s) for this situation must be available when the situation demands. Such procedures are termed daemons, and they were stored in a data structure separate from the normal procedures. The CAN_EXEC fields of these daemons look for the situations in which they are relevant.

A second modification for emergency procedures was substitute procedures. For example, in an engine-out emergency, the regular pre-landing procedure is replaced with a single engine pre-landing procedure. Substitute procedures were implemented by a pointer from the normal to the substitute.

EV'.LUATION AND RESULTS

The program was evaluated twice. The first time only normal procedures and normal flights were used. The program was then enhanced for the second evaluation, which used emergency and double emergency flights. The results for each evaluation are presented separately below.

Evaluation One

The program was first evaluated by developing the program on a derivation set of data and then running it unmodified on a validation set. The data was taken from normal flights and normal segments of emergency flights from the experiment of [Rouse, Rouse, and Hammer, 1982]. Flights were assigned randomly to derivation and validation data sets. As stated earlier, the objective was for the program to identify all of the errors found by Rouse and Rouse [1983].

The derivation data contained eight errors; as shown in Table 1 the program was able to locate seven of them positively and give an ambiguous indication of the eighth. This one error was omission of the cruise procedure and was originally located by examination of verbal transcripts. From the aircraft data recorded during the flight, the following can be determined. Of the three steps in the cruise procedure, the cowl flaps were definitely closed, the mixture might have been leaned (the necessary change might not have been enough for the computer to record), and the reduction in engine power was probably done, although one sensor reading required for the program to determine this was not available. It may be that the pilot executed the procedure without using its display or performing the callouts.

The validation data contained twelve errors. Eight errors were detected outright. Two er.ors were missed because a step was done incompletely and out-of-order. The program is designed to catch either of these errors individually; however, if both kinds of error are present in one step, the program will categorize it as done incompletely. Of the remaining two errors, one was turning a switch on, then off, then on, which was its intended state. The program simply checked off the step that required the switch to be on. At that time, the program did not test for conditions to be maintained. For the second evaluation, this shortcoming was fixed by the allowed field. The step that corresponds to, say, a switch change also ALLOWS that switch to change. When the step is checked off (i.e., DONE), its ALLOW field will no longer be checked. Since no other step will ALLOW the change, it would be detected if it occurs

The remaining error was an irrelevant action that would have been detected had it happened during an identified phase of the flight. Unfortunately, it happened between phases. Ideally, phases should overlap slightly so that the program has some phase to test the action. In the second evaluation, this shortcoming was fixed by having the ending of one phase force the next phase to begin.

The two types of error caught by the program are the following. One additional omitted procedure (besides the one mentioned earlier) was detected. Nine inappropriate actions were detected; most of these were activation of lights, etc. that were inappropriate at the times they occurred. Two instances were detected of lowering the landing gear at airspeeds higher than the maximum. Three instances were detected of not setting a control to the proper point. This included landing with partial instead of full flaps and not fully testing the ailerons before takeoff.

It might be expected that the program would find new errors that had been missed in the earlier investigation. It did not. While the program did turn up several cases of steps out-of-order, they were not really errors. For example, the step of retracting the flaps required so much time that the following step -- a discrete switch change -- could be completed while waiting for the flaps to retract. No new, substantial errors were found by the program.

Evaluation Two

The same methodology of derivation and validation data was used in the second evaluation. The results are shown in Table 2.

The one error the program did not detect was execution of two procedures when only one was needed (normal pre-landing and single engine pre-landing). The only detectable difference between these two is a single step -- the setting of the cowl flaps. At the time the procedure was invoked, the cowl flaps were in the position (one-half) that a step of the procedure requested they be. This step was immediately made DONE. Later, the pilot closed the flaps. The program, due to a simple bug in an ALLOW field, accepted this change. Eventually, the single engine pre-landing procedure was finished. The pilot then went through the normal pre-landing checklist, which resulted in no changes save for a different cowl flap setting. This change was detected as incorrect. If the simple bug were corrected, the program would not accept the first cowl flap change.

In addition to errors, the program identified several anomalies in pilot behavior. The most frequent was steps executed out-of-order. Expert opinion of these specific situations was that no error occurred. For example, the landing lights, navigation lights, and rotating beacon may be shut off in any order (once the propellers have stopped spinning) even though the procedure lists a specific order for them to be done.

These anomalies could be used for two kinds of improvements. The first would be to improve the program. In the above example, it would be better to express the ordering requirements semantically (e.g., engine off precedes beacon off) rather than by ordering. The second improvement could be to the procedures themselves. For example, flaps may not be extended above certain airspeeds. This restriction is not written in the Aztec procedures even though a similar restriction is written for landing gear, which is the step preceding flap extension. Such inadequacies could be found by coding the procedures in a program.

CONCLUSIONS

A model of procedure execution has been implemented in a computer program. It was tested on aircraft simulator data and was able to find practically all of the already known errors plus locate some new ones. While this serves as a practical test of the methodology, the implications of its aiding ability are more significant.

Using the model as an aid would have two benefits. The first, and most obvious, would be to detect and eliminate a great number of procedural errors. Perhaps surprisingly, this improvement comes with no additional pilot workload. A correctly functioning procedural aid would not need to communicate with the pilot except when an error was made.

The second benefit of the model would be display control. The latest generation aircraft are fitted with electronic displays that presumably could or do display procedures. The computer model of procedure execution could well be used to select and control displays, which might also result in an additional reduction in pilot workload.

ACKNOWLEDGMENT

The assistance of William B. Johnson, Sandra H. Rouse, and William B. Rouse is gratefully acknowledged.

This research was supported by the National Aeronautics and Space Administration under Ames Grant No. 2-123.

REFERENCES

Backus, J., "Can programming be liberated from the von Neumann style," Comm. ACM, Vol. 21, No. 8, 1978.

Bartlett, F. C., Remembering, Cambridge: London, 1932.

Buchanan, B. G., Sutherland, B., and Feigenbaum, E. A., "Heuristic Dendral: A program for generating explanatory hypotheses in organic chemistry," Machine Intelligence, Vol. 4, American Elsevier: New York, 1969.

Colley, R. W., "A transition control system for procedure prompting," Transactions of the American Nuclear Society, Vol 41, 1982.

Goldstein, I. P. and Grimson, E., "Annotated production systems: A model for skill acquisition," Proc. Fifth Inter. Joint Conf. Artificial Intelligence, 1977.

Goodstein, L. P., "Procedures for the operator -- Their role and support," IWG/NPPCI Specialists' Meeting on Procedure and Systems for Assisting an Operating during Normal and Anomalous Nuclear Power Plant Operation Situations, Garching, Federal Republic of Germany, December 1979.

Hammer, J. M. and Rouse, W. B., "Design of an intelligent computer-aided cockpit," Proceedings of the 1982 International Conference on Cybernetics and Society, pp. 449-453, Seattle, October 1982.

Minsky, M., "A framework for representing knowledge," in The Psychology of Computer Vision, P. H. Winston, ed., McGraw-Hill: New York, 1975.

Monan, W. P., "Distraction -- A human factor in air carrier hazard events," NASA Aviation Safety Reporting System, Ninth Quarterly Report, Moffett Field, CA: NASA-Ames Research Center, June 1979.

Norman, D. A., "Categorization of action slips," Psychological Review, Vol. 88, No. 1, 1981.

Rasmussen, J., "What can be learned from human error reports," in Proceedings of the NATO Conference on Changes in the Nature and Quality of Working Life, Thessalonica, Greece, 1979.

Rouse, S. H. and Rouse, W. B., "Computer-based manuals for procedural information," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10, No. 8, August 1980.

Rouse, S. H., Rouse, W. B., and Hammer, J. M., "Design and evaluation of an onboard computer-based information system for aircraft," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 4, July/August 1982.

Rouse, W. B. and Rouse, S. H., "Analysis and classification of human error," *IEEE Transactions on Systems, Man, and Cybernetics*, to appear, 1983.

Schank, R., and Abelson, R., *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum: Hillsdale, NJ, 1977.

Seeman, S. E., Colley, R. W., and Stratton, R. C., "Optimization of the man-machine interface for lmfbrs," American Nuclear Society International Topical Meeting on LMFBR Safety and Related Design and Operational Aspects, Lyon, France, July 1982.

Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN*, American Elsevier: New York, 1976.

van Eekhout, J. M., and Rouse, W. B., "Human errors in detection, diagnosis, and compensation for failures in the engine control room of a supertanker," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 12, December 1981.

Waterman, D. A., and Hayes-Roth, F., *Pattern-Directed Inference Systems*, Academic Press: New York, 1978.

Wesson, R. B., "Planning in the world of the air traffic controller," *Proc. Fifth Inter. Joint Conf. Artificial Intelligence*, Cambridge, MA, 1977.

Wiener, E. L. and Curry, R. E., "Flight-deck automation: promises and problems," Ergonomics, Vol. 23, No. 10, 1980.

1.	Mixture controls	full rich
2.	Propellers	full high rpm
3.	Throttles	set 2300 rpm
4.	Propellers	exercise 300 rpm max drop
5.	Propellers	feather check 500 rpm max drop
6.	Magnetos	check
7.		175 rpm max drop
8.		50 rpm max differential
9.	Engine gauges	check
10.		oil pressure
11.		oil temperature
12.		cylinder head temperature
13.		ammeter
14.		vacuum
15.	Throttles	set 1000 rpm

Figure 1. Engine Run-up Procedure

```
SCRIPT              = record
    NAME            : string;
    CAN_EXEC        : rule;
    DONE_EXEC       : rule;
    ABORT_EXEC      : rule;
    STATE           : [UNSTARTED,IN_PROGRESS,DONE,ABORTED];
    ALLOWED,
    DISALLOWED,
    OP_ALLOWED,
    INOP_ALLOWED,
    OP_DISALLOWED,
    INOP_DISALLOWED  : set of signal;
    COMP             : array[1..30] of Uscript;
    PARENT           : Uscript;
    SUB              : Uscript;
    end;
```

Figure 2.   Script fields.

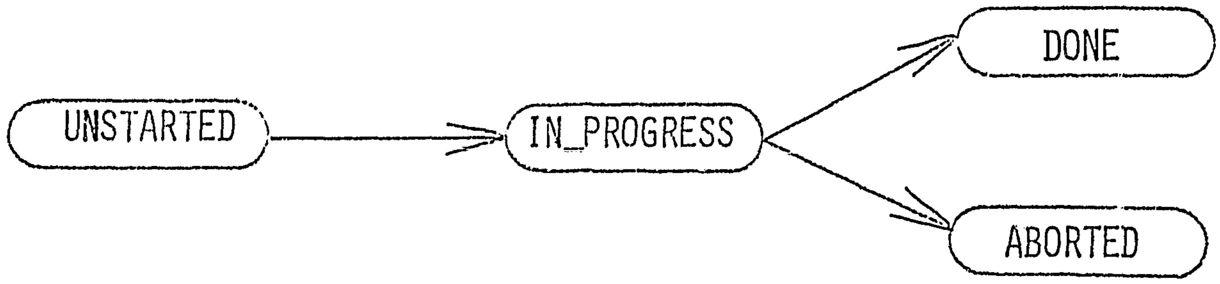


Figure 3. STATE transitions

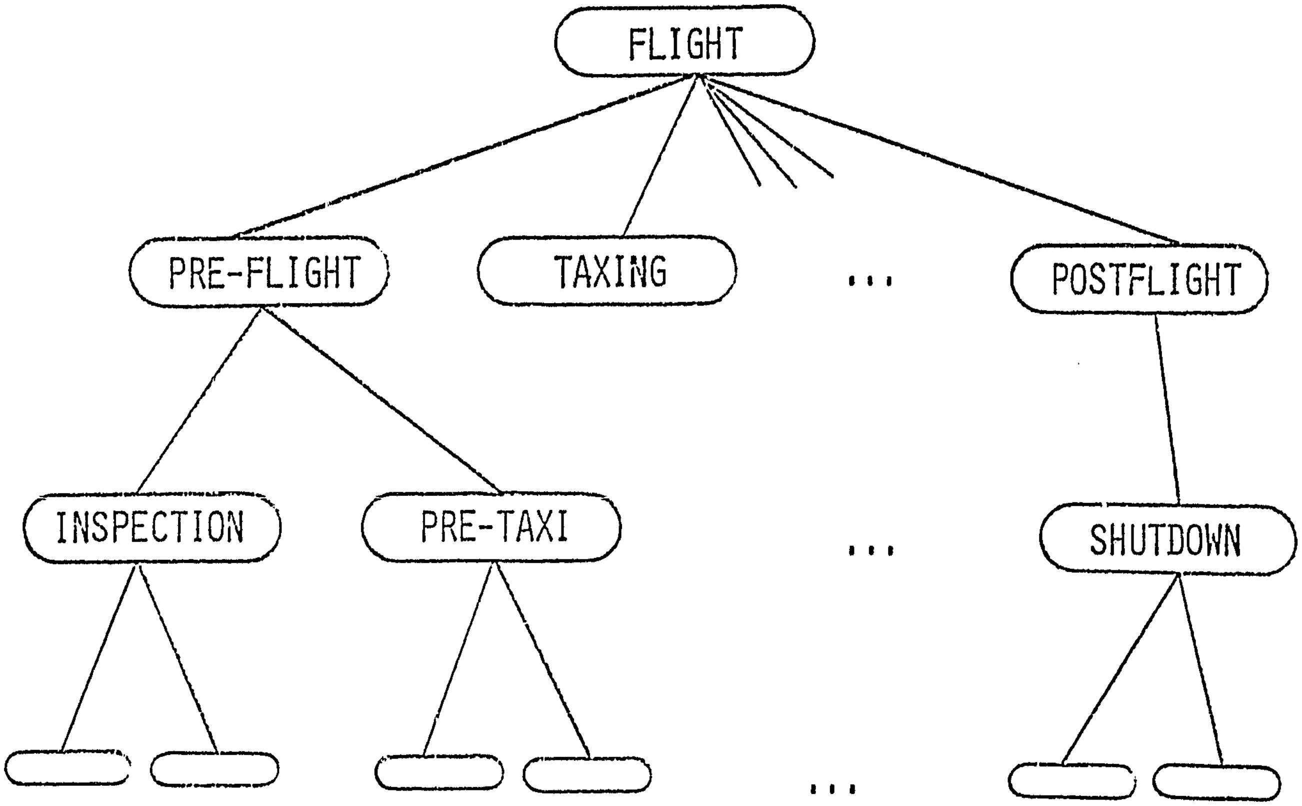


Figure 4. Hierarchy of steps, procedures, phases, and FLIGHT.

	Refound	Missed	Undetectable	New
Derivation	7	0	1	0
Validation	8	4	0	0

Table 1. Normal flight error analysis.

Refound errors were found by Rouse, Rouse, and Hammer and by the prograr. Missed errors were found by the original investigators but not by the program. Undetectable errors were found by the original investigators using source of data (i.e., cockpit tape recordings) that were not available to the program. New errors were found by the program but not by the original investigators.

	Refound	Missed	Undetectable	New
Derivation	6	0	4	3
Validation	9	1	4	5

Table 2. Emergency flight error analysis.

THE EFFECTS OF TYPE OF KNOWLEDGE UPON HUMAN PROBLEM SOLVING IN A PROCESS CONTROL TASK

Nancy M. Morris and William B. Rouse

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The question of what the operator of a dynamic system needs to know was investigated in an experiment using PLANT, a generic simulation of a dynamic production process. Knowledge of PLANT was manipulated via different types of instruction, so that four different groups were created: 1) Minimal instructions only; 2) Minimal instructions + guidelines for operation (Procedures); 3) Minimal instructions + dynamic relationships (Principles); 4) Minimal instructions + Procedures + Principles. Subjects controlled PLANT in a variety of situations which required maintaining production while also diagnosing familiar and unfamiliar failures. Despite the fact that these manipulations resulted in differences in subjects' knowledge as assessed via a written test at the end of the experiment, instructions had no effect upon achievement of the primary goal of production, or upon subjects' ability to diagnose unfamiliar failures. However, those groups receiving Procedures controlled the system in a more stable manner. Possible reasons for the failure to find an effect of Principles are presented, and the implications of these results for operator training and aiding are discussed.

INTRODUCTION

The role of operators of engineering systems, such as aircraft, ships, or process plants, has changed greatly in recent years and continues to change. Much of this change has been precipitated by advances in automatic control of systems. As the responsibility for control is shifted to computers, the operator becomes less a controller and more a monitor and, if necessary, a problem solver [1]. As a result, the operator of an automatically controlled system is called upon to exercise quite different skills from those required of the operator of a manually controlled system. Beyond some minimal level, psychomotor ability becomes less essential, and greater emphasis is placed upon the use of cognitive skills such as reasoning, pattern matching, and problem solving.

Realizing this, a variety of individuals concerned with system design and operator training have argued that one should "consider the cognitive processes of the operator" when dealing with design and training issues (e.g., [2], [3]]). Few people would dispute this idea, because the assertion that the operator's needs and capabilities should be considered seems to be a reasonable one. However, further development of the concept as stated here is required if it is to be practically useful.

From a theoretical viewpoint, theorists and researchers in the fields of psychology and artificial intelligence as well as within the domain of process control have discussed human cognition in a variety of problem situations. A number of models of reasoning and decision making have been offered, employing such concepts as schemas, scripts, heuristics, etc. (e.g., [4], [5], [6]). The general opinion is that, when faced with a problem, the human uses some understanding mechanisms which govern the situation in making decisions.

A construct which appears in many writings is that of the "mental model" of the process (e.g., [7], [8], [9], [10], [11], [12]). Although, unfortunately, the term has sometimes been employed rather loosely, the mental model has generally been used as a representation of knowledge of a system and its relationship with the environment. A number of functions have been attributed to the mental model, including guiding information seeking [11], [13], [14], aiding in pattern recognition [15], [16], and anticipating future system states [17].

One of the most articulate discussions of the relationships between mental models and problem solving in operation of engineering systems has been provided by Rasmussen [18]. In ordinary, familiar circumstances, the human operator appears to rely upon available heuristics and rules of operation. In other words, the operator's behavior is rule-based. However, in unusual situations for which rules do not apply, the human operator must reason at a knowledge-based level, using an understanding of the functioning of the system to determine an appropriate course of action. Thus, different mental models may be more or less appropriate, depending upon the situation.

From a practical perspective, the idea of considering operator cognitive processes and the notion that multi-level reasoning may be required have attracted the interest of those concerned with system design and the training of operators [19]. Practitioners have found, however, that the manner in which system designs and training programs should be structured so as to incorporate these ideas is not at all straightforward. For example, it has been suggested that one should strive to support the operator's reasoning and decision making process by providing information that enhances the operator's model [16], [20]. Yet, translating this suggestion into a specific course of action is not an easy task.

Speculation has been directed at the nature of the mental model associated with good performance. As a result of this speculation, it has been assumed both implicitly and explicitly that an important part of the mental model is a representation of the dynamics of the system. Some educators have further stated that such a representation (i.e., a "thorough understanding of the dynamics of the system") is a requisite if the operator is to be effective (e.g., [21]). Based upon this assumption, training programs may be aimed at providing the operator with the appropriate mental model, usually via instruction in the theory upon which the system is based and perhaps some experience with simulators. Often the further assumption that such instruction will lead to satisfactory performance is made.

Unfortunately, although these approaches may be intuitively appealing, there appears to be little in the way of empirical support to guide the practitioner's efforts. For example, there is lttle or no conclusive evidence that providing operators with information about theoretical aspects of system functioning enables them to be better operators. In fact, in research in which subjects were given instruction in the theoretical basis of system functioning there was no apparent advantage to having been given such information [9], [22], [23], [24]. It is quite possible that being able to control the system is not directly related to an explicit knowledge of system dynamics. Alternatively, it is conceivable that effective control behavior may be related to having an understanding of system dynamics, but that this understanding may be in the form of a "process feel" and may not be obtained via verbal instruction. At any rate, in spite of the lack of support for the practice, there is continued emphasis on instructing operators in the theoretical basis of system functioning.

The experiment reported in this paper was designed to investigate the question of what the operator of a dynamic system needs to know in order to be effective. In particular, the value of two different types of knowledge--knowledge of how to control the system, and knowledge of how the system works--was explored. The general approach was to manipulate system-relevant knowledge via instructions, and examine the effects of this knowledge upon performance.

A PROCESS CONTROL TASK

This research was conducted in the context of PLANT, a computer-driven generic simulation of a dynamic production process. A graphic display for a sample PLANT problem is shown in Figure 1, and the information display which accompanies the graphic display is shown in Figure 2. A general description of PLANT is presented here. Interested readers are referred to [25] for further details about the simulation.

Referring to Figure 1, in this system there are nine tanks, some of which are currently connected by open valves (represented by lines between tanks). Fluid enters the PLANT system at the left and exits at the right as finished product. In general, the PLANT operator's task is to supervise the flow of fluid through the series of tanks interconnected by pumps, valves, and pipes so as to produce an unspecified product. The operator may open and close valves, adjust system input and output, check flows between tanks, and order repairs of various PLANT components, in order to achieve the primary goal of maximizing production.

Each operator action, such as opening a valve or adjusting input, requires one time unit or iteration. PLANT is not updated automatically in real time, but rather is at steady-state between commands and is thus self-paced. Although it is possible for PLANT to run in a forced-paced mode and periodically update automatically (e.g., once every four seconds), the decision was made to employ the self-paced mode of updating because of the long response times characteristic of real processes.

As in real systems, although maximizing production is the primary goal of PLANT operation. the "physical" limitations of the system (such as tank or valve capacity or reliability of system components) require that the PLANT operator be concerned with secondary goals as well. Among these secondary goals are stabilization of the system, and detection, diagnosis, and compensation for system failures. Stability is required because of the dynamic characteristics of the system* and the fact that PLANT valves do not have infinite capacity. Should the operator fail to maintain stability, the PLANT safety system intervenes in order to protect the system from damage due to unsafe operating practices. The safety system operates by automatically closing valves (i.e., "tripping" them) and/or stopping system input or output if flows or fluid levels exceed desired ranges.

Possible PLANT failures include valve failures, pump failures, tank ruptures, and failure of the safety system. Valve and pump failures are fairly common, and involve a stoppage of flow between connected tanks. While flow is stopped, the display remains unchanged and, therefore, the failed valve or pump appears to be working. Detection and diagnosis of a valve or pump failure may be accomplished by noting a difference in observed and expected fluid levels in tanks, and checking flows through the suspected valve(s). Repair involves sending a "repair crew" to the site of the failure for a period of 5-10

*Each pair of connected tanks is modeled as a second-order system with rate of flow and its derivative as state variables and transition matrix determined by pipe and tank cross-sectional areas, pipe lengths. and fluid characteristics. See [26] for a derivation of the state equations.

iterations.

Tank ruptures and failure of the safety system are extremely rare by design, and may occur only once during a subject's experience with the system. As a result, these failures provide a means for studying operator problem solving in unfamiliar situations. A tank rupture must be inferred from noting a loss of resources from the system, and occupies the repair crew for 15 iterations, during which the tank is drained and "patched".

The nature of the failure of the safety system failure is much less predictable due to the range of possible safety system actions; it may be manifest by a number of different symptoms, and may be intermittent. For example. failure of the safety system could result in arbitrary "tripping" that should not be difficult to detect if one understands the way in which the safety system works. Thus, detection and diagnosis of a safety system failure requires that the operator have some knowledge of the functioning of the safety system and the underlying dynamics of the process, because safety system actions are directly related to PLANT dynamics. During repair, the safety system is deactivated for 20 iterations and the operator is responsible for PLANT safety.

With respect to the PLANT environment, it is possible to identify different types of knowledge about PLANT which the operator might have. At a minimum, he might know that he is controlling a process, his goal is to maximize production, and that various control options are available. At another level, the PLANT operator may know "what to do" in certain situations--i.e., he may have a set of procedures or rules which, when followed, enable adequate control of the system. Finally, it is possible for the operator to have a knowledge of the way in which PLANT "works", including an understanding of the underlying process dynamics and relationships between components.

In the research described in the following section, an attempt was made to "create" operators with these different types of knowledge by providing naive subjects with differing instructions. These operators were then placed in familiar and unfamiliar situations in order to provide them opportunities to use the information they were given. During the planning and conduct of this research, the following outcomes were expected. First, it was anticipated that those operators with a set of procedures for controlling PLANT would be better in ordinary or familiar situations than those without such information. Second, it was predicted that those persons with an understanding of PLANT dynamics and principles would be better able to deal with unfamiliar situations.

METHOD

Subjects

Junior and senior undergraduates at Georgia Institute of Technology served as paid volunteer subjects. All 32 were industrial and systems engineering majors, and had completed courses in physics, dynamics, and higher level mathematics.

It is important to note here that, although the use of students as subjects is often considered to compromise credibility in applied research. this subject population was well-suited to the questions at hand. This is due to the fact that operators in many systems (e.g., nuclear power plants) are required to complete a training program which is technically equivalent to that required for a bachelor's degree in engineering. Therefore, it is argued that these students had educational backgrounds comparable to actual operator trainees in some domains.

Experimental Materials

Four sets of written instructions relevant to PLANT were used in the experiment: Minimal instructions, Principles, Procedures, and Relationships Between Principles and Procedures. The format for the first three was similar, in that each consisted of text interspersed with "self-test" questions and accompanied by 1-2 page summaries of important concepts. The fourth set, Relationships, differed, as it was designed to be inserted into Procedures for an experimental group which was instructed using both Procedures and Principles. These instructions were designed to represent the types of knowledge about PLANT discussed earlier. (The complete sets of instructional materials appear in Morris' thesis [1].)

Minimal instructions were directed at what questions: what kind of system is it, what is the goal of operation, what can happen, etc. As such, Minimal instructions consisted of an introduction to the concept of a process plant, and a discussion of the goals of PLANT operation, operational constraints, possible malfunctions, and command options available. Self-test questions in the Minimal instructions were directed at insuring an understanding of the basics of PLANT operation (such as opening valves and adjusting input and output) and the nature of the PLANT safety system and possible PLANT malfunctions.

Procedures told the PLANT operator how the system should be controlled, in both general and more specific terms. First, there were three heuristics useful for general control of PLANT (e.g., "keep all valves open"). The Procedures also included a set of six more specific sequences of control actions (i.e., procedures in the formal sense) appropriate for use in a number of undesirable PLANT states (e.g., "output column too low"). These "specific sequences" were not as specific as the procedures used in aviation, but were more like "guidelines", discussing appropriate types of control actions rather than specific commands to be entered. The majority of the self-test questions required the subjects to determine which procedure was applicable in a depicted PLANT state (i.e., "Which procedure would you choose in this situation?").

These procedures were the product of numerous discussions between the authors. each of whom had considerable experience in controlling PLANT and had developed his/her own strategy for doing so. Procedures were evaluated for their "reasonableness" by actually using them to control the process; in instances where alternative procedures had been generated, the sequence of steps leading to the best performance (i.e., the most production and fewest valve trips) was selected.*

Principles included a presentation of an approximation of the state equations governing PLANT dynamics, and a verbal interpretation of the equations in terms of observable dynamic relationships. In short, the Principles indirectly contained information as to why PLANT should be controlled in a certain manner. In writing the Principles, an effort was made to make them as meaningful and relevant to PLANT operation as possible. Discussion of abstract theory was avoided, and mathematical expressions were always limited to simple algebraic expressions and accompanied by a discussion of their meaning and importance to PLANT functioning. For example. the instructions stated that the PLANT was "sluggish", that flows tended to "oscillate over time", and that input into a tank was "shared" by the valves leading from it. Self-test questions required the subject to apply the written information to the solution of problems (e.g., "If tank B had a level of 75 and tank F had a level of 63 when valve BF was opened, what would be your estimate of the initial flow rate for valve BF?").

Relationships Between Principles and Procedures were more directly related to the "whys" of PLANT operation. In Relationships, the rationale behind the information in the

*Throughout this paper, reference is made both to the set of procedural instructions and to operational procedures found in these instructions. To avoid confusion. references to the instruction set begin with an upper-case letter (i.e., Procedures), whereas "procedures" refers to specific sequences of steps found in the procedural instructions.

Procedures was presented in terms of concepts discussed in the Principles. Generally, subjects were informed, "You should (do this) because (the PLANT works this way)". As noted earlier, Relationships was inserted in Procedures for an experimental group which was instructed using both Procedures and Principles.

Two multiple-choice tests of the information in the instructions were also used. Test 1 contained 22 items, all related to information in the Minimal instructions. Test 2 consisted of 54 items, with approximately one third devoted to each of the major types of instruction (i.e., Minimal, Principles, and Procedures). Minimal questions on Test 2 were virtually identical to those on Test 1, with minor modifications. When creating procedural and principle questions, an effort was made to avoid asking questions which would be impossible to answer correctly without having been explicitly told the answer in instructions. For example, alternative answers often consisted of a range of numbers rather than specific values.

Experimental Method

Subjects served in a total of 12 sessions each, with the average length of each session being approximately 60 to 75 minutes. With the exception of sessions 10 and 12 (which were counterbalanced), the order of presentation of PLANT production runs was identical for all subjects. The first eight sessions were training sessions, in which subjects received written and oral instructions and controlled PLANT in a variety of situations for varying lengths of time. Material presented in instructions was repeatedly reviewed during training sessions, and all of subjects' questions were answered, if possible. in a manner appropriate to a particular subject's experimental condition.

The last four sessions were experimental sessions, and were identical in terms of initial PLANT configuration and length of production run. Sessions 9 and 11 were "familiar" runs, in that all failures which occurred were failures which the subjects had experienced before (i.e., valve and pump failures). Sessions 10 and 12 were "unfamiliar" runs, each involving a malfunction which had been discussed in instructional materials but which had never occurred in a subject's experience (i.e., tank rupture and safety system failure). The type of unfamiliar failure which occurred was counterbalanced across subjects and within instructional groups (described later). No instructions from the experimenter were provided during the last four sessions, and no questions from subjects were answered.

All subjects were presented with the Minimal instructions at the beginning of session 1, and were allowed to read them with the understanding that they would always have access to written materials when controlling PLANT. Following an oral review of the instructions with the experimenter, they were allowed to control PLANT for approximately one hour. During their first production run, they were encouraged to try all commands to make sure they understood how they worked.

Session 2 consisted of a brief review of commands and another one-hour production run. Test 1 was administered at the end of session 2. Since it was intended primarily as a vehicle for discussion, all correct and incorrect answers were discussed with subjects and important points were emphasized. Sessions 3 through 7 were "problem" runs, with subjects assuming control of the PLANT in a variety of unstable situations. These problems were created by the experimenter, and represented situations for which specific procedures were applicable. Sessions 8 through 12 were "normal" runs once more; as in sessions 1 and 2, no problems existed when the subject began controlling the PLANT. Test 2 was administered at the end of session 12.

Differentiation of experimental groups began in session 3. At the beginning of session 3, two groups of eight subjects each (groups B and D) were given Principles, and a third group (group C) was given Procedures. The remaining eight subjects (group A) were given no further written instructions. At the beginning of session 5, subjects in group D were also given Procedures, with Relationships Between Principles and Procedures inserted at the appropriate point.

To summarize, group A received Minimal instructions; group B received Minimal instructions and Principles; group C received Minimal instructions and Procedures; group D received all instructions. These four groups may be viewed as cells in a 2 x 2 factorial design, with each group receiving Procedures or no Procedures, and Principles or no Principles.

A number of measures of subjects' performance were recorded. In addition to the obvious performance measure of production, several intermediate measures were noted as indications of how "elegantly" subjects achieved their goal. Among these were the number of automatic valve trips, number of limit alarms (i.e., tank levels too high or too low), number of valves open per iteration, number of observations made prior to repairing a failure, variability of fluid levels both within and between columns, and frequencies of various commands.

RESULTS

Analysis of variance was used as the primary statistical tool for data analysis. Performance measures were used as dependent variables in three-way analyses with two between-groups factors (Principles and Procedures) and one within-groups factor or repeated measure (session). The following results are presented to provide an overview of the experimental findings. A more in-depth analysis of the results of this research may be found in [1].

When production achieved was used as the dependent variable in the analysis, there was no effect of either Procedures or Principles. The interaction also failed to reach significance. Of all the other performance measures, there were three which revealed significant differences related to instructions. These were the average number of automatic valve trips, average number of valves open at any point in time, and variance of fluid levels (i.e., tank heights) within the system.

All of the significant effects upon these variables were those of Procedures. Subjects provided Procedures (i.e., groups C and D) generally experienced fewer automatic valve trips (.94 vs. .66 per iteration, *p* = .0343), kept more valves open (15.79 vs. 14.58, *p* = .0074), and had less variance in tank heights (15.92 vs. 21.59, *p* = .0251) than did those subjects who did not receive Procedures (i.e., groups A and B). None of the main effects of Principles nor any of the Principles x Procedures interactions reached significance.

With regard to the unfamiliar failures, there was no difference in groups' ability to detect and repair the tank rupture or safety system failure. Only one person (from group D) did not repair the tank rupture, and approximately half in each instruction group repaired the safety system. Subjects were classified according to whether or not they repaired the failure of the safety system and the analysis of variance was repeated. (This classification is denoted by "fix-nofix" in the following discussion.) When differences in the variables noted above were analyzed in this manner, the following significant effects were noted.

First, those subjects who were able to determine that the safety system had failed generally produced more, regardless of session, than did those who did not make an appropriate diagnosis (321.3 vs. 298.7 units per iteration, *p* = .0303). Furthermore, "fixers" generally had fewer automatic valve trips (.68 vs. .94 per iteration, *p* = .0100), more valves open (15.64 vs. 14.68, *p* = .0128), and less variance in tank heights (15.92 vs. 21.59, p = .0001).

With respect to two of these variables, trips and height variance, the interaction of Procedures and fix-nofix was also significant (p = .0031 and p = .0061, respectively). Analysis of the simple main effects of these interactions revealed that the differences were among those subjects who did not repair the safety system. In other words, those persons who repaired the safety system were equivalent in terms of trips and height variance, regardless of whether or not they had been given Procedures. Among those persons who did not repair the safety system, however, those people who were not given Procedures had more valve trips (1.30 vs. 0.65) and height variance (28.32 vs. 16.15) than those who received Procedures.

Differences in performance on Test 2 were also identified via analysis of variance. When overall scores were compared, there were significant main effects both of Procedures and Principles (p = .0008 and p = .001, respectively). Groups receiving Procedures scored higher than those receiving no Procedures (80.44% vs. 70.94%), and Principles groups scored higher than those not receiving Principles (80.09% vs. 71.30%). The interaction of Procedures and Principles was not statistically significant.

Comparing scores on test sections (i.e., questions related to Minimal instructions, Procedures, and Principles), the interactions of Procedures x section (p = .0128) and Principles x section (p = .0008) were significant. Analysis of simple main effects revealed that subjects receiving Procedures answered more procedural questions correctly than those who did not receive Procedures (82.53% vs. 61.33%), and subjects given Principles correctly answered more questions related to system dynamics (72.13% vs. 48.07%).

Correlations between all dependent measures were computed, and a subset of these correlations may be found in Table 1. The results of the analyses presented earlier clearly demonstrate the existence of some strong relationships between variables. These correlations are offered as a mechanism for integrating the more detailed results into an overall picture, which is discussed in the following section.

DISCUSSION AND CONCLUSIONS

Interpretation of Results

There are three observations which may be made relative to the information presented in Table 1. First, the significant correlations between production, trips, number of open valves, and variance in tank heights are noteworthy because they provide support for the information found in Procedures. The main thrusts of these guidelines were aimed at keeping all valves open and controlling differences in tank heights. Judging from the relationships of height variance, etc. to overall production, these emphases were well-founded. The point is necessarily made because it is unreasonable to expect operators to follow rules which are not appropriate.

Second, the high correlations between number of valve trips, number of valves open. and variance in tank heights reflect characteristics of PLANT and provide justification for the treatment of these variables as alternative measures of a single construct. Thus, a "stable" PLANT is one in which most valves are open, there are few valve trips, and there is little variance in tank heights. The concept of PLANT stability is utilized in the following paragraphs when differences in control performance are discussed.

Third, perhaps the most important observation to be made from an examination of Table 1 is that the relationship between PLANT performance and Test 2 performance was not very strong. The highest correlation between PLANT production and any measure of Test 2 performance was .19, which was not significant. Of all the correlations between Test 2 and PLANT performance. only the relationship between number of open valves and score on the procedural section of Test 2 achieved significance.

Between group variations on Test 2 indicate that manipulation of instructions relative to PLANT was at least moderately successful in establishing different groups with respect to PLANT-relevant knowledge. In fact, the pattern of test results obtained is exactly as one might predict would occur if the manipulation were successful. It is also interesting to note that, since the interaction of Principles and Procedures was not significant, the effect of providing more than one set of instructions was approximately additive.

In contrast to the results on Test 2, instructions were not as clearly reflected in PLANT performance. For example. instructions had no effect upon how much subjects were able to produce. Regardless of instructions, groups were able to achieve comparable production scores. Although production was comparable across groups, those groups receiving Procedures (groups C and D) controlled PLANT in a more stable manner than did the groups without Procedures (groups A and B). The provision of Principles did not seem to improve subjects' control behavior under normal circumstances.

Variations in instructions had no effect upon whether or not a subject was able to correctly diagnose the unfamiliar failure of the safety system. Judging from the analysis of the Procedures x fix-nofix interaction, a stable system was apparently a necessary prerequisite to finding this malfunction. This is not surprising, since there would be a greater contrast between "normal" and "abnormal" in such a system. However, it is also apparent that having a stable system was not a sufficient condition for the location of the safety system failure. Procedures enabled subjects to have a more stable system, but only half of those subjects receiving Procedures repaired the safety system.

Restatement of Experimental Hypotheses

Now, consider the results of this experiment in light of the experimental hypotheses stated earlier. To reiterate, the first hypothesis was that those groups receiving Procedures (i.e., groups C and D) would be better at controlling PLANT in ordinary circumstances than those not provided Procedures (i.e., groups A and B). The data obtained in this research support this hypothesis. Although there were no differences between groups in overall production achieved, subjects in groups C and D generally controlled PLANT in a more stable manner, and were more consistent with each other with respect to most dependent measures. This evidence indicates (to no great surprise) that proceduralization may indeed be a means of providing operators with an effective strategy, and thus supports the common practice of providing operators with procedures.

The second hypothesis was that persons with an understanding of the dynamics of PLANT as described in Principles (i.e., groups B and D, or at least group D) would perform better in unusual circumstances in which available procedures were not applicable. The results reported here provide absolutely no support for this hypothesis. As reported earlier, only one person failed to repair the unfamiliar tank rupture, and approximately half of the subjects in <u>each</u> instruction group repaired the failed safety system. In retrospect, all subjects had been told in the Minimal instructions how to detect a tank rupture, so this failure to note a difference between groups in repair of the tank rupture is not too surprising; however, the pattern of results obtained with the safety system failure was not expected, and is difficult to explain.

The provision of Principles did <u>not</u> insure that subjects would be able to deal with the unfamiliar safety system failure. Neither did Principles appear to be useful in ordinary situations, as group B was no better than group A in controlling PLANT. In light of the performance on Test 2, it may be stated that this does not reflect a failure on the part of subjects to learn the material. Nor does it appear that this failure to find an effect may be attributed merely to failure to achieve the traditionally accepted significance level of .05. In all cases, measured differences due to an effect of Principles were small, and the probabilities of these differences being due to chance were quite large.

<u>Why not Principles</u>?

There are two questions which immediately come to mind when considering the failure to find support for the second hypothesis. The first is this: Why did Principles fail to help? It is necessary to address this question because of prevailing opinion as to the value of such a knowledge--the Principles <u>should</u> have helped. In fact, this attitude is so firmly held that some may even be led to discount the results reported here, because "everyone knows that you need to understand how a system works in order to control it".

In considering this question of why the provision of Principles did not lead to better performance, it is important to note that these results do not appear to represent an isolated case. Rather, they are in agreement with the results of other research in which knowledge of theory was found to have little or no relationship to task performance [9], [22], [23], [24], [27]. In fact, a survey of relevant literature failed to reveal any reports in which a statistically significant advantage of such knowledge was reported, although many authors stated or implied that there was such an advantage.

One approach to explaining these results might be to argue that the effects of knowledge of theoretical principles may be indirect and subtle, and thus not directly measurable. Indeed, a number of more subtle effects seem feasible, though a detailed examination of this data fails to support them. For example, a general understanding of the functioning of a system may serve as a frame of reference from which procedures may be more meaningful and better understood. Understanding how the system works may have a motivational effect upon operators. Although such knowledge may not be useful to a group of operators as a whole, some individuals may find this information extremely useful.

An additional explanation for this consistent failure to find an advandage of theoretical instruction may be in terms of different types of knowledge. The results of this research suggest that knowledge of a system may be represented in more than one form, and that any given person's knowledge may consist of multiple representations. Thus, knowledge of "facts", as measured by a verbal test, and knowledge of how to control a system, as manifest by adequate control performance, may not be strongly related and may be embodied in different forms and thus expressed in different ways. The low correlations between Test 2 scores and PLANT performance measures are consistent with this interpretation.

If this is the case, then the impact of Principles may have been minimal because the information was not in a form that was directly usable by subjects (i.e., was not directly related to what they should be able to do. as opposed to what they should know). Rather, in order to apply the information appropriately, the operator first had to go through a deductive process. Either people did not attempt to do so, or did try but could not determine an appropriate course of action. In the absence of successful reasoning, the Principles could not be useful.

Alternatives to Principles

The second question which arises when considering these results is this: If telling operators how the system works does not insure that they will be able to deal with unanticipated events, then what can be done to provide such assurance? This reflects a pressing need in industry, because it is precisely for the purpose of handling unforeseen situations that human operators are employed. Accordingly, an attempt will be made to address this issue here.

It is appropriate to recall the concept of multiple levels of reasoning discussed earlier. People commonly engage in rule-based behavior when controlling familiar systems under normal conditions, but should resort to knowledge-based behavior in unusual circumstances, using an understanding of the way the system works to determine what should be done. Therefore, if a person has a knowledge base sufficient to support knowledge-based reasoning, this information should be used in unfamiliar situations. Although this seems to be a reasonable description of what should occur, the indications from this research are that this describes the ideal and not what actually takes place. As we have seen, knowledge and opportunity do not guarantee that people will engage in knowledge-based reasoning and reach an appropriate conclusion.

It seems that certain conditions must be met for a person to solve an unfamiliar problem successfully. First, he or she must have an adequate knowledge base. Second, it must be apparent that available rules do not apply and that reasoning about the problem is required. Third, the person must be able to use the information in the knowledge base appropriately to reach a conclusion.

The nature of this "adequate" knowledge base was the primary question pursued in this research. and the partial answer obtained was "less than one might suppose". Some subjects from groups A and C found the safety system failure, and were generally quite good at controlling PLANT, yet could not answer questions on Test 2 about PLANT functioning. While it cannot be stated that these persons had no ideas of how PLANT works, it can be said that their knowledge of PLANT was at least less detailed than the information contained in Principles. Therefore, it appears that the importance of a detailed theoretical knowledge of a system to an operator's control behavior has been overemphasized in training, and this emphasis should be reduced.

Therefore, it may be necessary to provide the operator some assistance at the time of the unanticipated event, possibly online. One form of assistance could be to adequately inform the operator that an unusual condition existed. Other authors have indicated that it might also be necessary to help him to pinpoint the location of the problem. Finally, it could be necessary to guide the operator in his reasoning process, to increase the likelihood that an appropriate conclusion will be reached. Research in the areas of decision making and decision aiding is moving in the direction espoused in this paragraph [28]. However, any existing operator support systems of the type envisioned here are mainly in the conceptual stage and little evaluative data is available.

Summary

In summary, the question of what an operator needs to know is extremely important to those responsible for operator training. Traditionally, operators have been required to learn a great deal about the theoretical aspects of system functioning, in the hopes of insuring that they can deal with unanticipated events. Available research evidence suggest that this emphasis on the importance of theoretical knowledge of the system is disproportionate to the actual value of such knowledge, and that more attention should be devoted to providing operators assistance during abnormal conditions. In other words, less emphasis should be placed on answering the question of "What does the operator need to know?" and more on the questions of "What should operators be able to do?" and "How can we help them to use the knowledge they have?"

ACKNOWLEDGEMENT

This research was supported by the National Aeronautics and Space Administration under Ames Grant No. NAG 2-123.

REFERENCES

[1] N. M. Morris, "Human problem solving in process control." Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, Report No. 83-2, 1983.

[2] D. A. Norman, "Human cognition and skills." Proceedings of the Workshop on Cognitive Modeling of Nuclear Plant Control Room Operators, pp. 92-106, 1982.

[3] T. B. Sheridan, "Cognitive models and computer aids for nuclear plant control room operators," Proceedings of the Workshop on Cognitive Modeling of Nuclear Plant Control Room Operators, pp. 52-62, 1982.

[4] W. B. Rouse and R. M. Hunt, "Human problem solving performance in fault diagnosis tasks." In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. 1). Greenwich, Connecticut: JAI Press, 1983.

[5] D. E. Rumelhart and A. A. Abrahamson, "A model of analogical reasoning," Cognitive Psychology, vol. 5, pp. 1-28, 1973.

[6] R. C. Schank and R. P. Abelson, Scripts, Plans, Goals, and Understanding. Hillsdale. N. J.: Lawrence Erlbaum, 1977.

[7] S. N. Anyakora and F. P. Lees. "Detection of instrument malfunction by the process operator," Chemical Engineer, vol. 264, p. 304, 1972.

[8] C. L. Borgman, "Mental models: ways of looking at a system," ASIS Bulletin, vol. 9, pp. 38-39, 1982.

[9] H. Kragt and J. A. Landeweerd, "Mental skills and process control." In E. Edwards and F. P. Lees (Eds.), The Human Operator in Process Control. London: Taylor and Francis, 1974.

[10] D. E. Kieras and S. Bovair, "The role of a mental model in learning to operate a device." Tuscon, Arizona: University of Arizona, Report No. 13, 1983.

[11] J. Rasmussen, "On the structure of knowledge--a morphology of mental models in a man-machine system context." Roskilde, Denmark: Riso National Laboratory, Riso-M-2192, 1979.

[12] H. Stassen and R. J. van der Veldt, "Human operator models: a useful tool in MMIF design?" Delft, The Netherlands: Delft University of Technology, Laboratory for Measurement and Control, 1982.

[13] R. D. Smallwood, "Internal models and the human instrument monitor," IEEE Transactions on Human Factors in Electronics, vol. HFE-8, pp. 181-187, 1967.

[14] J. Rasmussen, "Outlines of a hybrid model of the process plant operator." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[15] R. E. Curry and E. G. Gai, "Detection of random process failures by human monitors." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[16] J. Rasmussen, "Models of mental strategies in proces plant diagnosis." In J. Rasmussen and W. B. Rouse (Eds.), Human Detection and Diagnosis of System Failures. New York: Plenum, 1981.

[17] T. B. Sheridan, "Toward a general model of supervisory control." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[18] J. Rasmussen, "Skills, rules and knowledge; signals, signs. and symbols, and other distinctions in human performance models," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, pp. 257-266, 1983.

[19] R. E. Hall, J. R. Fragola and W. J. Luckas, Jr. (Eds.), The 1981 IEEE Standards Workshop on Human Factors and Nuclear Safety. New York: Institute of Electrical and Electronics Engineers, 1981.

[20] L. P. Goodstein, "Discriminative display support for process operators." In J. Rasmussen and W. B. Rouse (Eds.), Human Detection and Diagnosis of System Failures. New York: Plenum, 1981.

[21] R. J. Braune and S. R. Trollip, "Towards an internal model in pilot training," Aviation, Space, and Environmental Medicine, vol. 53, pp. 996-999, 1982.

[22] A. Shepherd, E. C. Marshall, A. Turner and K. D. Duncan, "Diagnosis of plant failures from a control panel: a comparison of three training methods." Ergonomics, vol. 20, pp. 347-361, 1977.

[23] E. R. F. W. Crossman and J. E. Cooke, "Manual control of slow-response systems." Paper presented at the International Congress on Human Factors in Electronics, Long Beach, California, 1962. Also in E. Edwards and F. P. Lees (Eds.), The Human Operator in Process Control. London: Taylor & Francis, 1974.

[24] B. W. Surgenor and J. D. McGeachy. "Validation for performance measurement in the task of fault management," Proceedings of the 27th Annual Meeting of the Human Factors Society, pp. 1058-1062, 1983.

[25] N. M. Morris, W. B. Rouse, and J. L. Fath, "Studies of human problem solving in process control: Evolution of an experimental simulation." Submitted for possible publication in IEEE Transactions on Systems, Man, and Cybernetics.

[26] J. L. Fath, "Design and evaluation of a process plant simulator." Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, Report No. 82-3. 1982.

[27] W. L. Williams, Jr. and P. G. Whitmore, Jr., "The development and use of a performance test as a basis for comparing technicians with and without field experience: the Nike Ajax maintenance technician." Washington, D. C.: George Washington University. Human Resources Research Office. Report No. 52, 1959.

[28] W. B. Rouse and S. H. Rouse, "A framework for research on adaptive decision aids." Norcross, GA: Search Technology, Inc., Report No. 83-1, 1983.

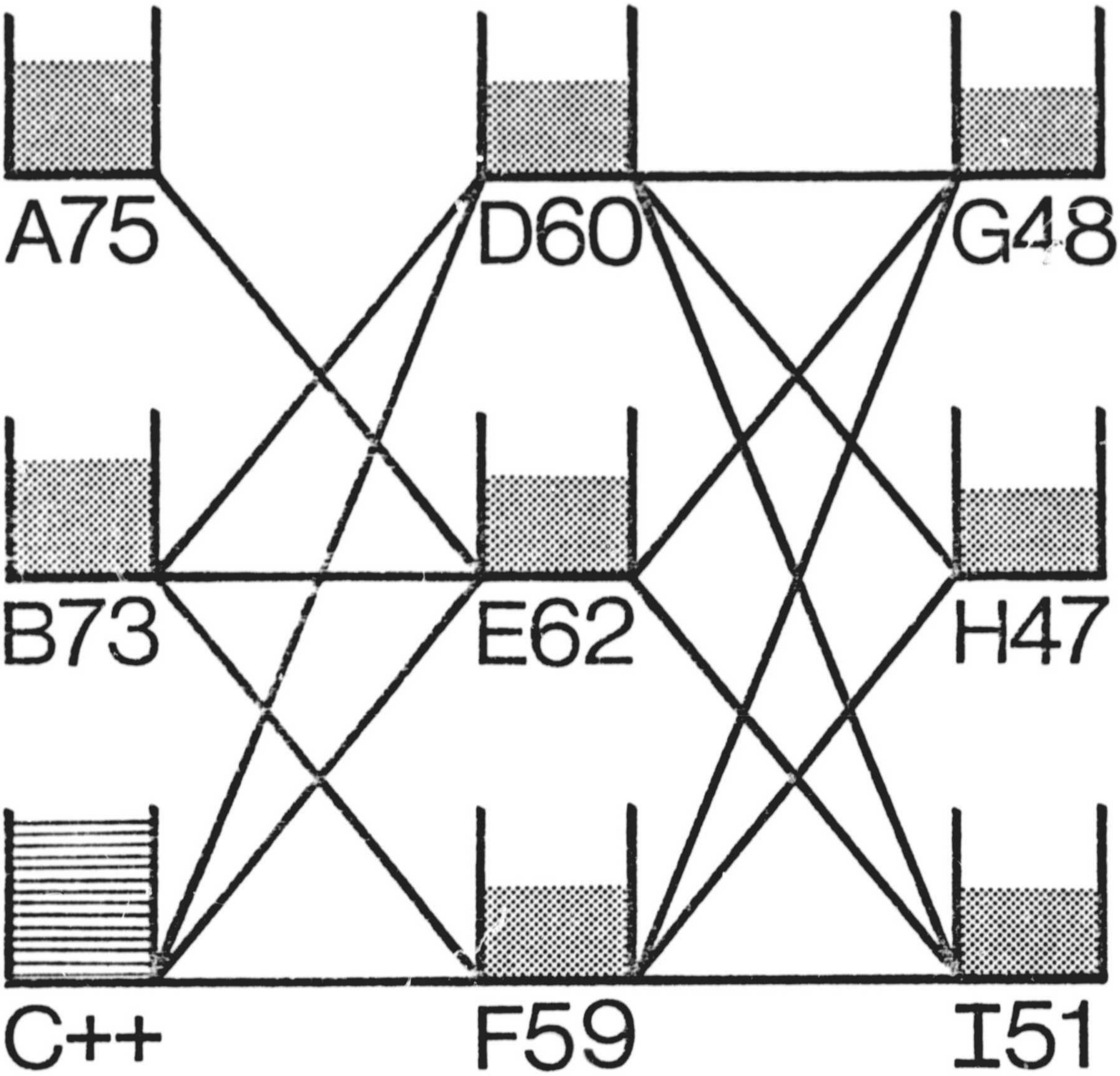
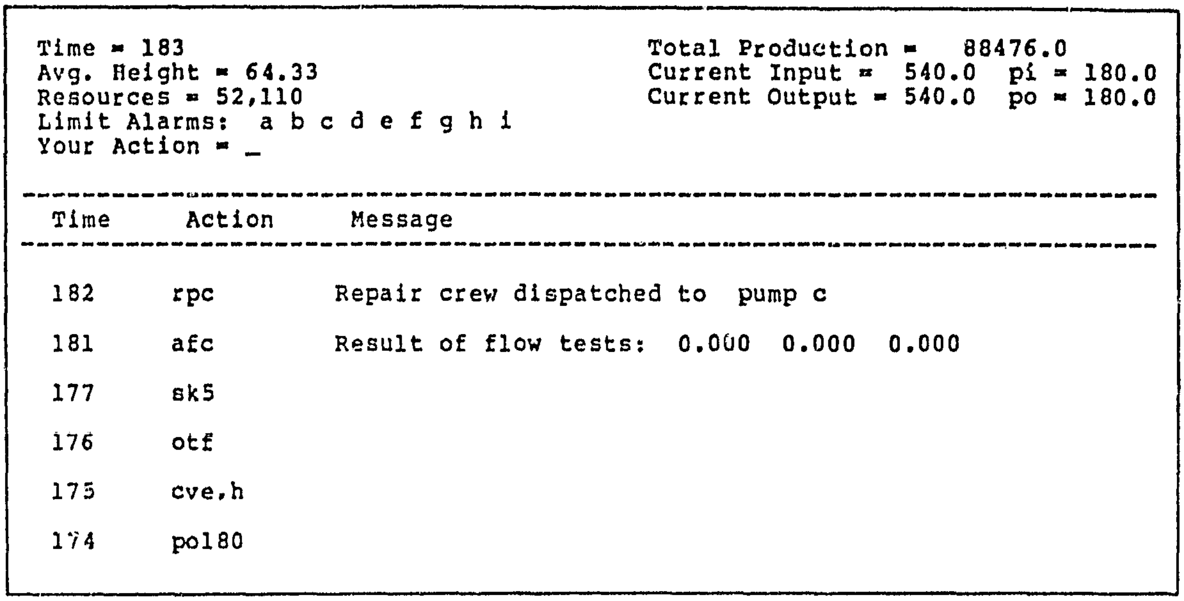


Figure 1. Sample graphic PLANT display.



```
Time = 183                                      Total Production =    88476.0
Avg. Height = 64.33                             Current Input =   540.0  pi = 180.0
Resources = 52,110                              Current Output =  540.0  po = 180.0
Limit Alarms:  a b c d e f g h i
Your Action = _

-----------------------------------------------------------------------------
 Time     Action     Message
-----------------------------------------------------------------------------

 182      rpc       Repair crew dispatched to  pump c

 181      afc       Result of flow tests:  0.000  0.000  0.000

 177      sk5

 176      otf

 175      cve,h

 174      po180
```

Figure 2. PLANT information display.

Table 1

Correlations Between Dependent Measures

	PROD[a]	TRIPS[b]	NOPEN[c]	VAR[d]	FIX[e]	TEST2[f]	SECT1[g]	SECT2
TRIPS	-.437*							
NOPEN	.673*	-.706*						
VAR	-.574*	.967*	-.768*					
FIX	-.429*	.141	-.234	.218				
TEST2	.191	-.200	.313	-.258	.107			
SECT1	-.021	.261	-.189	.268	-.100	.148		
SECT2	.190	-.238	.366*	-.292	.225	.860*	.040	
SECT3	.105	-.161	.157	-.197	-.056	.661*	-.022	.225

[a] PROD = average production/iteration.

[b] TRIPS = number of automatic valve trips/iteration.

[c] NOPEN = average number of valves open/iteration.

[d] VAR = variance of tank heights in PLANT.

[e] FIX = average time to diagnose valve and pump failures.

[f] TEST2 = overall score on Test 2.

[g] SECT1, SECT2, SECT3 = scores (% correct) on subsections of Test 2; SECT1 = minimal questions, SECT2 = procedural questions, SECT3 = principles questions.

*$p < .05$